UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/702,320 | 10/31/2000 | Laurence R. Simar, Jr. | TI-30559 | 9784 |

23494 7590 02/22/2005

TEXAS INSTRUMENTS INCORPORATED
P O BOX 655474, M/S 3999
DALLAS, TX 75265

| EXAMINER | |
|---|---|
| HUISMAN, DAVID J | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2183 | |

DATE MAILED: 02/22/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/702,320 | SIMAR, JR. ET AL. | |
| | Examiner | Art Unit | |
| | David J. Huisman | 2183 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *09 November 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1,4,7 and 9* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,4,7 and 9* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *13 January 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

## DETAILED ACTION

1. Claims 1, 4, 7, and 9 have been examined.

### *Papers Submitted*

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: RCE as received on 11/9/2004.

### *Claim Rejections - 35 USC § 112*

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claim 1 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

5. Claim 1 recites the limitation "an execute packet" in line 17. There is insufficient antecedent basis for this limitation in the claim as it is not clear whether or not applicant is referring to the execute packet of line 16.

### *Claim Rejections - 35 USC § 103*

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claim 1 is rejected under 35 U.S.C. 103(a) as being unpatentable over Simar et al., European Patent Application, EP 0855648A2 (as applied in the previous Office Action and herein referred to as Simar) in view of Hull et al., U.S. Patent No. 5,922,065 (as previously disclosed and herein referred to as Hull).

8. Referring to claim 1, Simar has taught a digital processing system having a microprocessor (Fig.1, component 11), wherein the microprocessor comprises:

a) fetch circuitry for fetching instruction fetch packets from sequential memory address locations, wherein each fetch packet contains a first plurality of fixed length instructions, each instruction including an instruction type and a predetermined p-bit, said p-bit having a first digital state indicating a next instruction is to execute in parallel with said instruction and a second digital state indicating a next instruction is to execute in a cycle after said instruction. See Fig.3 and page 3, lines 19-30.

b) a second plurality of functional units, each of the second plurality of functional units operable to execute a corresponding instruction in parallel with other functional units. See Fig.20, Fig.21, and page 4, lines 14-18. Note that execution units L2, S2, M2, and D2, for instance, are considered a second plurality of functional units which will execute L2-type, S2-type, M2-type, and D2-type instructions, respectively. And, these units will execute instructions in parallel with the L1, S1, M1, and D1, for instance, are considered other functional units which will execute L1-type, S1-type, M1-type, and D1-type instructions, respectively.

c) dispatch circuitry (Fig.20, component 10b) connected to said fetch circuitry and said second plurality of functional units operable to:

c1) select an execution packet from one or more fetch packets, wherein an execute packet varies in size and contains only a set of instructions that can be executed in parallel on the plurality of functional units, by scanning instructions from lower memory address locations to higher memory address locations and adding an instruction to said execute packet when said p-bit of a prior instruction has said first digital state until said p-bit of an instruction has said second digital state. See Fig.3, page 3, lines 19-30, and page 5, line 54, to page 6, line 7. Note that the fetch packet comprises 4 instructions (A, B, C, and D) and that different size execution packets are formed based on the values of the p-bits. See Fig.8-19. For example, in Fig.14, instructions are read from lower memory address to higher memory address (as shown in Fig.3), starting with instruction A. Instruction A is added to the execute packet and instruction A's p-bit is checked and determined to be 1. Therefore, this p-bit value indicates that the next instruction (instruction B), may execute in parallel with instruction A. As a result, instruction B is added to the execute packet. Instruction B's p-bit is 0, which means that instruction C must start a new execute packet since it may not execute in parallel with instruction B. The final execute packets are then shown in Fig.15 (note that instructions A and B are in the same packet and C is in its own packet).

c2) Simar has not taught selecting an execute packet from two fetch packets by scanning instructions from lower memory address locations to higher memory address locations beginning in a first fetch packet, adding an instruction to said execute packet when said p-bit of a prior instruction has said first digital state and continuing past an end of said first fetch packet to a beginning of a second fetch packet until said p-bit of an instruction

has said second digital state. However, Hull has taught the concept of executing, in parallel, instructions from multiple fetch packets based on a stop bit. See Fig.3 and Fig.4 and column 3, line 61, to column 4, line 19. Such a bit allows for the definition of inter-bundle (inter fetch packet) boundaries to be defined, which is an extremely valuable processor function. See column 4, lines 55-60. Plus, such combining of fetch packets would maximize processor efficiency. For example, looking at Fig.4, assume that the processor is to first execute a bundle having a template value of 1 and then execute a bundle having a template value of 2. The double lines 42 in the first bundle means that the slot 2 instruction cannot be executed with the slot 0 or slot 1 instruction due to an intra-packet boundary (based on dependencies). Therefore, without the combining of multiple fetch packets, in cycle 0, the slot 0 and slot 1 instructions will execute. In cycle 1, the slot 2 instruction will execute. On the other hand, with the combining of packets based on a stop bit indicating no inter-bundle boundaries, in cycle 0, the slot 0 and slot 1 instructions will execute. However, this time, in cycle 1, instead of just one instruction (slot 2 instruction) executing, the slot 2 instruction along with instructions from the second bundle will execute. Therefore, more instructions may be executed per cycle when fetch packets may be combined. As a result, it would have been obvious to one of ordinary skill in the art of the invention to modify Simar to include inter-packet combining as taught by Hull.

c3) dispatch each instruction of said selected execute packet to a functional unit corresponding to said instruction type of said instruction. See page 3, lines 45-47, of Simar. Note that instructions A and B are executed in parallel, and for them to be

executed they must be dispatched to the execution (functional) units in a manner similar to that shown in Fig.24. For instance, if instruction A is an ADD instruction and instruction B is a multiply instruction, then instruction A will be dispatched to the L1 execution unit, since the L1 unit is an arithmetic logic unit, and instruction B will be dispatched to the M1 execution unit since M1 is a multiplier unit (page 4, lines 14-16).

9. Referring to claim 7, Simar has taught a method of operating a digital system having a microprocessor (Fig.1, component 11), wherein the microprocessor has a plurality of functional units (Fig.24) for executing instructions in parallel, comprising the steps of:

a) storing fixed length instructions at sequential memory address locations, each instruction including an instruction type and a predetermined p-bit, said p-bit having a first digital state indicating a next instruction is to execute in parallel with said instruction and a second digital state indicating a next instruction is to execute in a cycle after said instruction. See Fig.3 and note that instructions A, B, C, and D, were stored at sequential memory locations x00, x01, x10, and x11. Also, note that in addition to each instruction inherently including an instruction type, each instruction also includes a p-bit with the aforementioned digital states. See Fig.3 and page 3, lines 19-30.

b) fetching a sequence of instruction fetch packets, wherein each fetch packet contains a first plurality of instructions. See Fig.24, packets 1710, 1720, 1730, and 1740. Note that the system will fetch a sequence of instruction packets that contain a first plurality of instructions (in this case the first plurality = 8).

c) scanning the p-bit of each instruction of each fetch packet from lowest memory address location to highest memory address location to determine an execute packet dependent on the p-

bits. See Fig.3, page 3, lines 19-30, and page 5, line 54, to page 6, line 7. Note that the fetch packet comprises 4 instructions (A, B, C, and D) and that different size execution packets are formed based on the values of the p-bits. See Fig.8-19. For example, in Fig.14, instructions are read from lower memory address to higher memory address (as shown in Fig.3), starting with instruction A. Instruction A is added to the execute packet and instruction A's p-bit is checked and determined to be 1. Therefore, this p-bit value indicates that the next instruction (instruction B), may execute in parallel with instruction A. As a result, instruction B is added to the execute packet. Instruction B's p-bit is 0, which means that instruction C must start a new execute packet since it may not execute in parallel with instruction B. The final execute packets are then shown in Fig.15 (note that instructions A and B are in the same packet and C is in its own packet).

d) Simar has not taught scanning the p-bit of each instruction of each fetch packet from lowest memory address location in a first memory fetch packet to highest memory address location in a second immediately following fetch packet to determine an execute packet dependent on the p-bits. However, Hull has taught the concept of executing, in parallel, instructions from multiple fetch packets based on scanning a value of a stop bit. See Fig.3 and Fig.4 and column 3, line 61, to column 4, line 19. Such a bit allows for the definition of inter-bundle (inter fetch packet) boundaries to be defined, which is an extremely valuable processor function. See column 4, lines 55-60. Plus, such combining of fetch packets would maximize processor efficiency. For example, looking at Fig.4, assume that the processor is to first execute a bundle having a template value of 1 and then execute a bundle having a template value of 2. The double lines 42 in the first bundle means that the slot 2 instruction cannot be executed with the slot 0 or slot 1

instruction due to an intra-packet boundary (based on dependencies). Therefore, without the combining of multiple fetch packets, in cycle 0, the slot 0 and slot 1 instructions will execute. In cycle 1, the slot 2 instruction will execute. On the other hand, with the combining of packets based on a stop bit indicating no inter-bundle boundaries, in cycle 0, the slot 0 and slot 1 instructions will execute. However, this time, in cycle 1, instead of just one instruction (slot 2 instruction) executing, the slot 2 instruction along with instructions from the second bundle will execute. Therefore, more instructions may be executed per cycle when fetch packets may be combined. As a result, it would have been obvious to one of ordinary skill in the art of the invention to modify Simar to include inter-packet combining as taught by Hull.

e) dispatching each instruction within the determined execute packet to one of a second plurality of execution units dependent upon an instruction type of the instruction. See page 3, lines 45-47, of Simar. Note that instructions A and B are executed in parallel, and for them to be executed they must be dispatched to the execution (functional) units in a manner similar to that shown in Fig.24. For instance, if instruction A is an addition-type instruction and instruction B is a multiply-type instruction, then instruction A will be dispatched to the L1 execution unit, since the L1 unit is an arithmetic logic unit, and instruction B will be dispatched to the M1 execution unit since M1 is a multiplier unit (page 4, lines 14-16).

10. Claims 4 and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Simar in view of Hull, as applied above, and further in view of Heishi et al., U.S. patent No. 6,324,639 (as applied in the previous Office Action and herein referred to as Heishi).

11. Referring to claim 4, Simar has taught a system as described in claim 1. Simar has not explicitly taught the specifics of claim 4. However, Heishi has taught that the dispatch circuitry comprises:

a) a first latch to hold said first plurality of instructions of a first packet, said first latch including a first plurality of sections, each section storing a corresponding one of said first plurality of instructions of said first fetch packet. See Fig.5A and note that the fetch packets come in bundles of three units. These three units may each be a short instruction as shown by pattern (g) in Fig.7 and Fig.18. The first latch (Fig.8, component 221) also has three sections, each section receiving a corresponding instruction of that packet. For instance, each section of the latch would receive one short instruction from the bundle. This is shown in Fig.9B.

b) a second latch to hold said first plurality of instructions of a second fetch packet immediately following said first fetch packet, said second latch including a first plurality of sections, each section storing a corresponding one of said first plurality of instructions of said second fetch packet. See Fig.8, component 222. Note the second latch has three sections like the first latch 221. Again, each section has the ability to receive one of the three short instructions (if the packet contains three short instructions).

c) a first plurality of multiplexers (Fig.8, components 224), each multiplexer having a first input receiving an entire instruction from a predetermined section of said first latch (for instance, mux 224a receives an instruction from slot A0), a second input receiving an entire instruction from a corresponding section of said second latch (mux 224a receives an instruction from slot B0), a control input (column 13, lines 13-18, and note that component 223 in Fig.8 controls the muxes since the muxes must inherently be controlled) and an output (note the muxes output to an

instruction register), each multiplexer selecting at said output said entire instruction from said section of said first latch, said entire instruction from said section of said second latch, or no instruction, dependent upon said control input (see fig.9A-9F, and note that at times instructions exist within the latches but are not selected by the muxes, and sometimes instructions are selected from the first or second latches). It should be realized that if the instructions are short instructions, then the entire instruction will be selected by the multiplexer. In addition, it should be further realized from Fig.9A-9F and Fig.10A-10E, that the first multiplexer 224a will always select the oldest (or smallest) unit number. Therefore, the section that is selects is predetermined because the multiplexer must select from the section which holds the oldest unit (smallest unit number).

d) a dispatch control circuit connected to said first latch, said second latch, and said plurality of multiplexers, said dispatch control circuit receiving said predetermined p-bit from each instruction of said first latch and each instruction of said second latch for control of said plurality of multiplexers via said control inputs according to the execute packets determined by said p-bits. See Fig.11 and note that the issuing control circuitry 31 is connected to the instruction register which in turn is coupled to the multiplexers. Component 31 actually receives a p-bit from each instruction, which is the most significant bit of each instruction. See Fig.11, Fig.6A-F, and column 9, line 50, to column 10, line 3. Basically, these bits determine which instructions are sent by the multiplexers to be decoded and issued. Therefore, they control the multiplexers.

e) a cross point circuitry connected to said outputs of said plurality of multiplexers for dispatching said instructions at said output of said multiplexers to a functional unit corresponding to said instruction type of each instruction. See Fig.11, and note that the decoders

are considered a cross point circuitry since after the instructions are decoded, they are dispatched to the execution units 41. And the decoders get instructions from the instruction register which received the instructions from the multiplexers (see Fig.9). Therefore, the cross-point circuitry is coupled to the outputs of the multiplexers in that ultimately, the instructions which are outputted by the multiplexers are received by the decoders.

Heishi has taught that such circuitry allows for accommodation of variable length instructions. See Fig.15 and Fig.16, and note that words from multiple slots in the latches may be combined to form a larger instruction, such as one of the format shown in Fig.6D-F. A person of ordinary skill in the art would've recognized that larger size opcodes, displacement values, and immediate values may be implemented when implementing variable length instructions, thereby increasing the user's flexibility in that the user would not be restricted to such small numbers. For instance, see Fig.6B and 6E. Note that the user would be allowed to have a much larger immediate value. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Simar to include such circuitry so that more flexibility can be achieved.

12. Referring to claim 9, Simar in view of Hull has taught a method as described in claim 7. Simar has further taught that said step of determining an execute packet boundary dependent upon the p-bits includes:

a) storing each instruction of said first fetch packet and said second fetch packet. See Fig.24, and note that multiple fetch packets are brought within the system for processing. Simar has not explicitly taught that each instruction of the first and second fetch packets is actually stored in corresponding sections of a first and second latch, respectively. However, Heishi has taught the concept of having two latches and storing two separate fetch packets in them. See Fig.8,

components 221 and 222. Also, see Fig.9B and 9C and note that a first packet comprising units 1, 2, and 3 are stored in the first latch, and units 4, 5, and 6 are stored in a second latch. These units can correspond to short instructions as shown in Fig.18. A person of ordinary skill in the art would have recognized that, in general, it is more efficient to store fetch packets internally (in a latch, cache, etc.) so that a slow main memory access is not required. Having these latches will also mask the main memory access time. For instance, when one packet is being dispatched, no other packet can be dispatched. Therefore, a next packet can be retrieved from memory as opposed to just waiting until the dispatch finishes and then performing the main memory access. By overlapping (pipelining) the main memory access and dispatching, the system becomes more efficient in that when the dispatch does finish, instead of going to slow memory to get the next packet, the packet is retrieved from within the system which is much faster. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Simar in view of Heishi such that first and second fetch packets are store in first and second latches, respectively.

b) Simar in view of Heishi has further taught selecting an entire instruction from a predetermined section of said first latch, an entire instruction from a corresponding section of said second latch, or no instruction, dependent upon said p-bit from each instruction stored in said first latch and each instruction stored in said second latch. See Fig.24, for instance, and note that instructions are selected from the fetch packet, which would be stored in a latch, according to Heishi. This would also hold true for the second fetch packet (for instance, packet 1720 shown in Fig.24). And, if a p-bit indicates that a next instruction may not be executed in parallel, then that instruction will not be selected. It should be realized that if the instructions are short

instructions, then the entire instruction will be selected by the multiplexer. In addition, it should be further realized from Fig.9A-9F and Fig.10A-10E, that the first multiplexer 224a will always select the oldest (or smallest) unit number. Therefore, the section that is selects is predetermined because the multiplexer must select from the section which holds the oldest unit (smallest unit number).

### *Conclusion*

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
January 17, 2005

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100